Maintaining CloudForms Management Engine 3.x UI Performance

The CloudForms Management Engine (CFME) product is the successor product of the ManageIQ Enterprise Virtualization Management (EVM) Version 4 product.

Both of these products are highly dependent on a responsive database environment in order to maintain responsive User Interface (UI) interactions.

There are significant schema changes between the EVM V4 and CFME products which tend to reduce the need for periodic maintenance of the Virtual Management Data Base (VMDB) in the CFME environment, however there are still elements which will benefit from more frequent targeted table and index maintenance to extend the interval between full database reorganizations that might otherwise be necessary to maintain a responsive CFME environment. The use of the recommendations contained in this document will not adversely impact the operation of the CFME 3.0 operation but has not been evaluated for use in any other environment.

While this document references locations that exist within a CFME appliance, it is expected that these materials only be installed and activated on **one** appliance within a CFME domain, that appliance on which the VMDB is hosted and where the Postgresql server itself is executing. The initiation of the Postgresql 'psql' command line utility is used in such a way that it is assumed that no requirement for user-id or password is required based on the standard 'pg_hpa.conf' file installed with the CFME 3.0 release. If this is not the environment into which you propose to introduce these materials, it will be necessary for you to modify the use of the 'psql' command line to adapt to the requirements of your specific environment.

Maintaining well organized SQL tables and indices is a well-known and well documented requirement for database responsiveness. The intent of this database maintenance is to address the known effects of frequent updates to tables and to their associated indices by periodically compressing the base table and/or indices into the smallest amount of space needed to represent the table and index data in the filesystem on which that information is stored and to reduce the necessary memory requirements to keep that same information in memory where there is sufficient frequency of access to warrant such action by the database management system.

The postgresql.conf file provided with the CFME 3.0 installation is such that those files which are subject to the most frequent update are frequently and automatically processed by daemons that attempt to do much of this work. However, the current version of Postgresql (9.x) does not yet offer tools that do for the indices what these daemons do for tables. The materials referenced in this article are intended to address this issue.

From an operational perspective, the recommendation to periodically perform a full database reorganization (https://access.redhat.com/site/articles/312243, recommended backup and restore script "cfme_5.2_vmdb_backup_and_restore_scripts_20131206.tgz" found in https://access.redhat.com/site/

articles/531963) is one that generally isn't discovered until there is a problem with the UI response time for some set of UI requests. Generally these are requests that access either event timelines or performance history for some element within the VMDB. Organizing and performing this maintenance can be problematic.   This process requires exclusive access to the database with no read or write activity from any agent other than from that portion of the database management system which is performing the reorganization and this can run for long periods of time, on the order of several hours. Operationally, this means an outage to customers and may require scheduling downtime which becomes additionally extended as all appliances referencing the VMDB must be quiesced before the reorganization and then restarted after the reorganization completes. The reorganization itself may run for many hours resulting in an extended outage that many organizations may require to be scheduled during weekend or other non-peak usage periods.

The proposed maintenance actions contained in this document can be safely implemented so that they run fairly frequently and for that reason have fairly short durations (seconds to tens of seconds for hourly recommended actions) and provide for a more continuous database maintenance process with no externally noticeable impact to operational availability to end users. The proposed changes are grouped into several sets of **cron** jobs such that in very large or very active CFME environments, one job set can be scheduled on a more frequent basis (and in general are recommended for CFME production implementations of all sizes) while the others on a less frequent basis (daily to monthly) depending on customer specific factors.

The proposed changes are grouped into multiple (currently two) sets of **cron** jobs with names suggestive of the proposed frequency:
1. Periodic - intended to be schedule to run during non-peak hours during the weekend or possibly daily depending on duration and possible impacts in your environment. In this context the notion of "non-peak" is intended to be interpreted in the context of each installation's environment. So for some installations, daily at 1:00 AM might be appropriate, while for others weekly at 10:00 PM Saturday night might make better sense, and for yet others, monthly on the first Saturday of each month at 1 am where the day of the month is > 2 and < 10 or some other constraint. It should be understood that the longer the interval between execution, the longer each command will execute, but if scheduled during an off hour period this should not be of concern to anyone.
2. Hourly - intended to be scheduled hourly

As of Aug 12, 2014 I am providing a single revised set of scripts within the attachment 'cfme_vmdb_hourly_and_weekly_continuous_maint_scripts_2014-08-12.tgz'. This set of revised materials is substantially the same as the previous two scripts but is revised and includes two more files to make installing these cron jobs easier by copying one file to the /etc/cron.hourly directory and another to the /etc/cron.weekly directory of the vmdb owning appliance. Also, the outputs have been normalized so that all output from the scripts is directed to *.log files in the /var/www/miq/vmdb/log directory so that the files are under logrotate management and are automatically collected when appliance logs are collected using either the manual batch log collection or UI initiated log collection processes.

Download the 'cfme_vmdb_hourly_and_weekly_continuous_maint_scripts_2014-08-12.tgz' file into /var/www/ miq/vmdb and then decompress it. It will create a new directory named 'continuous_maint' containing the files to be executed. Copy the file 'zzz_hourly_vmdb_continuous_maint' from the new directory to /etc/cron.hourly

directory and the file 'zzz_weekly_vmdb_continuous_maint' to the /etc/cron.weekly directory and the cron jobs are installed